#18
4-19-04
DW

Attorney's Docket No.: 042390.P6447

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application for:

Michael Chow, et al.

Serial No.: 09/505,949

Filed: February 15, 2000

For: **METHOD AND APPARATUS FOR ACHIEVING ARCHITECTURAL CORRECTNESS IN A MULTI-MODE PROCESSOR PROVIDING FLOATING-POINT SUPPORT**

Examiner: Li, Aimee J.

Art Group: 2183

APPEAL BRIEF

Mail Stop Appeal Brief - Patent
Commissioner for Patents
P. O. 1450
Alexandria, VA 22313-1450

Dear Sir:

Applicants submit, in triplicate, the following Appeal Brief pursuant to 37 C.F.R. §1.192 for consideration by the Board of Patent Appeals and Interferences. Applicants also submit herewith a check in the amount of $330.00 to cover the cost of filing the opening brief as required by 37 C.F.R. § 1.17(f). Please charge any additional amount due or credit any overpayment to deposit Account No. 02-2666.

04/14/2004 EFLORES 00000114 09505949
01 FC:1402 330.00 OP

TABLE OF CONTENTS

I. REAL PARTY IN INTEREST

Michael Chow, Elango Ganesan, John William Phillips and Nazar Abbas Zaidi, the parties named in the caption, transferred their rights to that which is disclosed in the subject application through an assignment recorded on March 20, 2000 (010760/0706) in the patent application to Intel Corporation, of Santa Clara, California. Thus, as the owner at the time the brief is being filed, Intel Corporation, of Santa Clara, California is the real party in interest.

II. RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences which will affect or be affected by the outcome of this appeal.

III. STATUS OF CLAIMS

Claims 1-19 are pending and rejected in this application. Applicants hereby appeal the rejection of all pending claims.

IV. STATUS OF AMENDMENTS

The claims are amended in accordance with the Response Amendment filed on August 8, 2003, wherein Claim 18 was amended. The claim amendments requested in the Response Amendment filed on January 15, 2004 regarding Claims 10 and 19 were not entered.

V. SUMMARY OF THE INVENTION

The present invention describes a multi-mode processor to process instructions from a first instruction set architecture (ISA) having a first word size and to process instructions from a second ISA having a second word size where the second word size of the second ISA is different than the first word size of the first ISA. As described at page 7 of Applicants' specification,

> In one embodiment, a processor is capable of operating in two modes. The first and second modes are a 32 bit word ISA and a 64 bit word ISA, respectively. More specifically, the first mode is IA-32 mode, in which the processor emulates a 32 bit word Intel architecture (IA). . . . [T]he second mode is IA-64, which implements what is known as the IA-64 ISA. (pg. 7, lines 1-16.)

Accordingly, in one embodiment, the multi-mode processor provides backward compatibility or legacy support for a 32 bit word (legacy) ISA, as well as support for a 64–bit word (current) ISA. As described, the term "word size", as known to those skilled in the art, refers to the number of bits that a CPU can process at one time. In one embodiment, the processor includes floating point registers and floating point units, which are shared between a legacy ISA engine and a current ISA engine, as illustrated with reference to FIG. 1. In a computer's arithmetic logic unit (ALU), there sometimes exists input encodings, which the machine must interpret as

tokens, which require special handling. To provide support for such encodings, the current ISA includes a special FP encoding ("NaTVal"). When such a value is detected, the processor known value causes a floating point unit to ignore the requested operation and propagate the NaTVal as output, typically causing the processor to later request the data and/or operation non-speculatively when it is needed. (*See* pg. 8, lines 2-19.)

Unfortunately, the legacy ISA does not support NaTVal tokens. However, because the floating point registers and floating point units are shared between the legacy ISA engine and current ISA engine of the multi-mode processor, in one embodiment, as illustrated with reference to FIG. 2, preprocessing hardware 162 detects whether a NaTVal token (or other token and special values) is present in input operands. When a NaTVal token or other special token is detected by preprocessing hardware 166, depending on what mode the processor is in, values other than a true arithmetic result are prepared by post-processing hardware 166. (*See* pg. 11, lines 4-17.)

Operation of the multi-mode processor is illustrated with reference to FIG. 3. Representatively, preprocessing hardware analyzes input operands to detect which input operands must be interpreted as tokens instead of being fed to the arithmetic unit. Also, where a result is produced, post-processing hardware replaces the arithmetic result with the expected result given the special input tokens. However, the capability to inject the special result is turned on or off depending on whether the multi-mode processor is processing instructions from the legacy ISA or the current ISA which supports NaTVal tokens.

VI. ISSUES

The issues involved in this appeal are as follows:

Are Claims 1-4, 6, 8, 10-16 and 19 unpatentable under 35 U.S.C. §102(e) as being anticipated by Blomgren, U.S. Patent No. 5,884,057 ("Blomgren")?

Are Claims 9 and 17 unpatentable under 35 U.S.C. §103(a) as being unpatentable over Blomgren, U.S. Patent No. 5,884,057 ("Blomgren") in view of IEEE Standard for Binary Floating-Point Arithmetic ("IEEE")?

VII. GROUPING OF CLAIMS

Applicants submit that claims do not stand or fall together. Accordingly, Applicants group the claims as follows:

Group I	Claims 1, 2, 4, 5, 7, 8 and 18
Group II	Claim 3
Group III	Claim 6
Group IV	Claim 9
Group V	Claims 10-13, 15, 16, 19
Group VI	Claim 14
Group VII	Claim 17.

Applicants contend that all of the pending claims do not stand or fall together for the following reasons:

The claims in Group I specifically recite

> a first instruction set engine to process instructions from a first instruction set architecture (ISA) having a first word size and a second instruction set engine to process instructions from a second ISA having a second word size, the second word size being different than the first word size. (Emphasis added.)

The claims in Group II require preprocessing hardware and post-processing hardware to produce an output depending on a processor mode and whether an input includes a token.

The claims in Group III require that a token representing a "not a thing value" (NaTVal) defines an unsuccessful speculative load request.

The claims of Group IV require a 32 bit word ISA mode and a 64 bit word ISA mode.

The claims of Group V require pre-processing and post-processing to produce an output depending on a processor mode and whether an input includes a token.

The claims of Group VI require that a token representing a "not a thing value" (NaTVal) defines an unsuccessful speculative load request.

The claims of Group VII require a 32 bit word ISA mode and a 64 bit word ISA mode.

In addition, other limitations require the claims to be grouped as indicated. Applicants will argue why each of these groups of claims should be allowed.

VIII. ARGUMENT

A. Overview of the Invention and Cited References

1. Overview of Invention

The present invention is directed to a multi-mode processor. Distinctive features of the multi-mode processor include:

(i) a first instruction set engine to process instructions from a first instruction set architecture (ISA) having a first word size;

(ii) a second instruction set engine to process instructions from a second ISA having a second word size, the second word size being different than the first word size; and

(iii) preprocessing and post-processing to produce an output depending on a mode of the multi-mode processor and whether an input operand includes a token.

The multi-mode processor supports a legacy ISA having, for example, a 32 bit word size and a current ISA, having a 64 bit word size. An arithmetic logic unit (ALU) of the multi-mode processor interprets certain input encodings as tokens that require special handling by the ALU. To provide support for such encodings, the current ISA includes a special floating point

(FP) encoding ("NaTVal"). Unfortunately, the legacy ISA does not support NaTVal tokens. However, because the FP registers and FP units are shared between a legacy ISA engine and a current ISA engine of the multi-mode processor, multi-mode processor includes preprocessing and post-processing hardware.

As illustrated with reference to FIG. 2 of Applicant's specification, preprocessing hardware 162 detects whether a NaTVal token (or other special token and special values) are present in input operands. When a NaTVal token or other special token is detected by the preprocessing hardware 166, depending on what mode the processor is in, values other than the true arithmetic result are prepared by post-processing hardware 166. (*See* pg. 11, lines 4-17.) Accordingly, the capability to inject the special result is turned on or off, depending on whether the multi-mode processor is processing instructions from the legacy ISA or the current ISA, which supports NaTVal tokens.

2. Overview of Blomgren Reference

Blomgren teaches a mechanism for rapid reconfiguration of pipeline alignment between a pipeline optimized for RISC instructions and one optimized for CISC instructions with the use of muxes and mode registers. (Abstract, and also as depicted with reference to Blomgren's Fig. 2.) As indicated in Blomgren:

> The RISC and CISC instruction sets have independent encoding of instructions to opcodes. While both sets have ADD operations, the opcode number which encodes the ADD operation is different for the two instruction sets. In fact, the size and location of the opcode field in the instruction word is also different for the two instruction sets. Thus two instruction decoders are used for the two instruction sets – a RISC decoder 36 and a CISC decoder 32. (Col. 6, lines 37-44.) [Emphasis added.]

To this end, Blomgren describes various mechanisms for aligning the CISC and RISC pipelines, as depicted with reference to Figs. 2-3 and 5-7.

As illustrated with reference to FIG. 4:

> the CISC instruction decoder 32 detects these emulated instructions and signals from unknown opcode over line 40 to mode control logic 30. In response, the mode control logic 30 sets RISC bit-60 in register 38 and loads the instruction pointer with the address of the emulation routine in memory. Once the emulation routine is complete, an RISC instruction causes the mode register 38 to be reset to CISC mode and the instruction pointer updated to point to the following CISC instruction. The CISC program continues with the following instruction unaware that the instruction was emulated with RISC instructions. (*See* cols. 6-7 lines 61–10.) (Emphasis added.)

Accordingly, Blomgren teaches the emulation of complex CISC instructions using a plurality of RISC instructions, which are referred to as "emulated instructions".

> The instruction decoder 32 is responsible for detecting such instructions, which results in the setting of an RISC mode and loading of the emulation routine

in memory, such that following execution, the mode register 38 is reset to CISC mode and the instruction pointer is updated to point to the following CISC instruction. (Col. 7, lines 1-10.) (Emphasis added.)

In other words, Blomgren teaches the detection of an emulated CISC instruction and switches to a RISC mode when such an instruction is detected. The switch to the RISC mode results in the loading of various RISC instructions to perform the CISC instruction. Once completed, the mode is switched to the CISC mode to achieve rapid reconfiguration of pipeline alignment between a pipeline optimized for RISC instructions and a pipeline optimized for CISC instructions.

3. Overview of IEEE Reference

IEEE relates to an American National Standard (ANSI) for binary floating point arithmetic. IEEE describes the existence of the infinity input encodings. According to the standard described in IEEE, the infinity input encoding require special handling when received as an input to an arithmetic logic unit (ALU). Accordingly, in floating point systems which follow the standard described in IEEE, the value of infinity has a special encoding. The correct answer to the function ($\infty + x$) is infinity. However, if the encoding for infinity actually passed through to the adder in the ALU, a garbage value would result. (*See* pp. 9–10 of IEEE.) Accordingly, to produce the correct result requires the identification of one of the inputs as infinity and injection of infinity as the result.

B. Group I: Rejection of Claims 1, 2, 4, 5, 7, 8 and 18 as Anticipated by Blomgren

The Examiner rejected all pending claims, including Claim 1, 2, 4 and 8 of Group I under 35 U.S.C. §102(e) as being anticipated by Blomgren.

1. Errors of Law and Fact in the Rejection

Applicant respectfully asserts that the Examiner has failed to adequately set forth a *prima facie* rejection under 35 U.S.C. §102(b). "Anticipation requires the presence in a single prior art reference disclosure of each and every element of the claimed invention, *arranged as in the claim.*" *Lindemann Maschinenfabrik v. American Hoist & Derrick* ("Lindemann"), *730 F.2d 452, 1458 (Fed. Cir. 1994)*(emphasis added). Additionally, each and every element of the claim must be exactly disclosed in the anticipatory reference. *Titanium Metals Corp. of American v. Banner* ("Banner Titanium"), *778 F.2d 775, 777 (Fed. Cir. 1985).*

Although the Examiner has rejected Claims 1, 2, 4 and 8 as anticipated by Blomgren, the Examiner fails to show that Blomgren teaches the claimed subject matter.

According to the Examiner, at col. 6, lines 40-42, Blomgren teaches first and second instructions set engines to process instructions having a first word size and a second word size, the second word size being difference (sic) than the first word size. (*See* Final Office Action

mailed 11/04/03.) However, after having carefully reviewed the relevant portions of Blomgren cited by the Examiner, Applicants must respectfully disagree with the Examiner's contention.

Applicants submit that the Examiner is improperly equating the term "instruction size" with the term "word size", as required by independent Claims 1 and 18 of Group I. As defined by the Computer Dictionary Online, the term "word size" refers to the number of bits that a CPU can process at one time. (*See* Exhibit 1) Applicants submit that the number of bits a CPU can process at one time does not vary depending on whether the CPU is processing CISC instructions or RISC instructions.

Furthermore, Applicants submit that there is no suggestion as to any variation in the word sizes between the RISC and CISC instruction sets within either Blomgren or the references of record.

Specifically, as described in Blomgren:

> The RISC and CISC instruction sets have independent encoding of instructions to opcodes. While both sets have ADD operations, the opcode number which encodes the ADD operation is different for the two instruction sets. In fact, the size and location of the opcode field in the instruction word is also different for the two instruction sets. Thus two instruction decoders are used for the two instruction sets – a RISC decoder 36 and a CISC decoder 32. (Col. 6, lines 37-44.) [Emphasis added.]

Applicants submit that the cited passage implies that the instruction word size is the same for both the CISC and RISC instruction sets, and in accordance with the definition of "word size" provide above. Unless the CISC and RISC instruction words are the same size, Blomgren's indication of "the size and location of the opcode field in the instruction word being different for the two instruction sets" makes no sense. In other words, use of the definite article in this context requires a single instruction word size. Accordingly, one skilled in the art would not interpret Blomgren as teaching instruction set engines for processing instructions from ISAs having different word sizes.

Applicants submit that the entire description of Blomgren is devoid of any reference to providing processing for different ISAs having different ISA word sizes. However, the case law is quite clear in establishing that each and every element of the claim must be exactly disclosed in the anticipatory reference. Banner Titanium, *Id.* Hence, a *prima facie* case of anticipation of the claims over Blomgren has not been established and the rejection of Claims 1, 2, 4, 5, 7, 8 and 18 is therefore erroneous.

2. Specific Limitations Not Described in the Prior Art

Each of independent Claims 1 and 18 require:

(1) a first instruction set engine to process instructions from a first ISA having a first word size;

(2) a second instruction set engine to process instructions from a second ISA having a second word size, the first word size being different than the second word size. (Emphasis added.)

3. Explanation Why Such Limitations Render the Claims Unanticipated by the Prior Art

The Applicants claim a multi-mode processor that includes a first instruction set engine to process instructions from a first instruction set architecture ISA having a first word size. The multi-mode processor also includes a second instruction set engine to process instructions from a second ISA having a second word size, the second word size being different than the first word size. As described at pg. 7 of Applicants' specification:

> In one embodiment, a processor is capable of operating in two modes. The first and second modes are a 32 bit word ISA and a 64 bit word ISA, respectively. More specifically, the first mode is IA-32 mode, in which a processor emulates a 32 bit word Intel Architecture (IA) known as IA-32 ISA . . . [T]he second mode is a IA-64, which implements what is known as the IA-64 ISA. (pg. 7, lines 1-16.) (Emphasis added.)

Accordingly, in one embodiment, the multi-mode processor provides backward compatibility or legacy support for a 32 bit word (legacy ISA), as well as support for a 64 bit word (current) ISA.

In contrast, Blomgren teaches a mechanism for rapid reconfiguration of pipeline alignment between a pipeline optimized for RISC instructions and one optimized for CISC instructions, with the use of muxes and mode registers. (*See* Abstract and FIG. 2 of Blomgren.) To this end, Blomgren describes various mechanisms for aligning the CISC and RISC pipelines, as depicted with reference to FIGS. 2-7. Blomgren teaches the emulation of complex CISC instructions using a plurality of RISC instructions, which are referred to as emulated instructions. (*See* col. 7, lines 1-10.)

In other words, Blomgren teaches the detection of an emulated CISC instruction and switches to a RISC mode when such an instruction is detected. The switch to the RISC mode results in a loading of various RISC instructions to perform the CISC instruction to enable alignment between a pipeline optimized for RISC instructions and a pipeline optimized for CISC instructions.

Hence, Applicants submit that the entire description of Blomgren is devoid of any reference to providing processing for different ISAs having different ISA word sizes. Applicants submit that the cited passage above (*See* col. 6, lines 37-44 of Blomgren) implies that the

instruction word size is the same for both the CISC and RISC instruction sets, and in accordance with the definition of "word size" provide above.

Yet, in spite of the lack of any teaching toward processing of ISAs having a different word size, by improperly equating the term "instruction size" with the term "word size", the Examiner incorrectly finds a teaching within Blomgren to anticipate the claims of Group I. However, the case law is quite clear in establishing that each and every element of the claim must be exactly disclosed in the anticipatory reference. Banner Titanium, *Id.*

Therefore, a *prima facie* case of anticipation of the claims of Group I is not established and the rejection of Claims 1, 2, 4, 5, 7, 8 and 18 is erroneous and should be overturned. Accordingly, Applicants respectfully request that the §102(e) rejection of the Claims of Group I be overturned.

C. Group II: Rejection of Claim 3 As Anticipated by Blomgren

The Examiner rejected all pending claims, including Claim 3 of Group II under 35 U.S.C. §102(e) as being anticipated by Blomgren.

1. Errors of Law and Fact in the Rejection

The Examiner has made the same errors as previously described with respect to the rejected claims of Group I. In addition, the Examiner has failed to show that each and every element of Claim 3 of Group II is exactly disclosed by Blomgren. Banner Titanium. *Id.*

According to the Examiner, the use of a mode register to indicate the emulation of a CISC routine with RISC instructions, which results in the CISC instruction decoder 32 signaling an unknown opcode from line 40 to mode control logic 30 anticipates Claim 3 of Group II.

Hence, Blomgren teaches the emulation of complex CISC instructions using a plurality of RISC instructions, which are referred to as "emulated instructions".

> The instruction decoder 32 is responsible for detecting such instructions, which results in the setting of an RISC mode and loading of the emulation routine in memory, such that following execution, the mode register 38 is reset to CISC mode and the instruction pointer is updated to point to the following CISC instruction. (Col. 7, lines 1-10.) (Emphasis added.)

Applicants submit that the cited passage provides no reference to pre-processing hardware to determine whether a token is received as an input and post-processing hardware to perform a token specific operation, as required by Claim 3 of Group II. Hence, a *prima facie* case of anticipation of the claims of Group II has not been established and the rejection of Claim 3 is therefore erroneous.

2. Specific Limitations Not Described in the Prior Art

Claim 3 of Group II requires:

> pre-processing hardware to detect if a token exists in the input;
> an arithmetic unit responsive to the input and the mode identifier;
> and
> post-processing hardware to perform a token specific operation if a token exists in the input.

3. Explanation Why Such Limitations Render the Claims Unanticipated by the Prior Art

The Applicants claim a processor including pre-processing and post-processing hardware to process an input to render an arithmetic result if a processor is in a first mode and performing a token specific operation if a processor is in a second mode when a token is detected as an input. In contrast, Blomgren teaches a mechanism for rapid reconfiguration of pipeline alignment between a pipeline optimized for RISC instructions and one optimized for CISC instructions with the use of muxes and mode registers. (*See* Abstract and FIG. 2 of Blomgren.)

Blomgren teaches the emulation of complex CISC instructions using a plurality of RISC instructions, which are referred to as "emulated instructions". As described at col. 7, lines 1-10:

> The instruction decoder 32 is responsible for detecting such instructions, which results in the setting of a RISC mode in loading the emulation routine in memory, such that following execution, the mode register 38 is reset to CISC mode and the instruction pointer is updated to point to the following CISC instruction.

In other words, Blomgren teaches the detection of an emulated CISC instruction and switches to a RISC mode when such an instruction is detected. However, the cited passage above, as well as the entire text of Blomgren is devoid of any reference to detecting whether an input from a plurality of floating point registers includes a token.

Furthermore, Blomgren is devoid of any teachings of performing token specific processing by post-processing hardware if a token is received as an input. In contrast, Blomgren teaches a CISC instruction decoder that detects emulated instructions and signals the unknown opcode, which the Examiner equates to a token, to load pointers with the address of an emulation routine. (*See* cols. 6-7, lines 61-10.)

Accordingly, although Blomgren describes emulation of certain CISC routines with RISC instructions, the processing performed within Blomgren does not vary according to a processor mode as indicated by a mode identifier. Applicants respectfully submit that the detection of emulated instructions does not teach the detection of whether an input contains a token, as required by Claim 3 of Group II.

Therefore, a *prima facie* case of anticipation of the claims of Group II is not established and the rejection of Claim 3 should be overturned. Accordingly, Applicants respectfully request that the §102(e) rejection of Claim 3 of Group II be overturned.

D. Group III: Rejection of Claim 6 As Anticipated by Blomgren

The Examiner rejected all pending claims, including Claim 6 of Group III under 35 U.S.C. §102(e) as being anticipated by Blomgren.

1. Errors of Law and Fact in the Rejection

The Examiner has made the same errors as previously described with respect to the rejected claims of Groups I and II. In addition, the Examiner has failed to show that each and every element of Claim 6 of Group III is exactly disclosed by Blomgren. Banner Titanium. *Id.*

According to the Examiner, the use of a mode register to indicate the emulation of a CISC routine with RISC instructions, which results in the CISC instruction decoder 32 signaling an unknown opcode from line 40 to mode control logic 30 anticipates Claim 6 of Group III.

Namely, Blomgren teaches the emulation of complex CISC instructions using a plurality of RISC instructions, which are referred to as "emulated instructions".

> The instruction decoder 32 is responsible for detecting such instructions, which results in the setting of an RISC mode and loading of the emulation routine in memory, such that following execution, the mode register 38 is reset to CISC mode and the instruction pointer is updated to point to the following CISC instruction. (Col. 7, lines 1-10.) (Emphasis added.)

Applicants submit that the cited passage provides no reference to a token that represents a "not a thing value" (NaTVal) defines an unsuccessful speculative load request, as required by Claim 6 of Group III. Hence, a *prima facie* case of anticipation of Claim 6 of Group III has not been established and the rejection of Claim 6 is therefore erroneous.

2. Specific Limitations Not Described in the Prior Art

Claim 6 of Group III requires:

> wherein the token represents a "not a thing value" (NaTVal) that defines an unsuccessful speculative load request.

3. Explanation Why Such Limitations Render the Claims Unanticipated by the Prior Art

The Applicants claim a processor including pre-processing and post-processing hardware to perform a token specific operation if a processor is in a second mode, when a token is detected, wherein a token representing a "not a thing value" (NaTVal) defines an unsuccessful speculative load request. In contrast, Blomgren teaches a mechanism for rapid reconfiguration of pipeline alignment between a pipeline optimized for RISC instructions and one optimized for CISC instructions with the use of muxes and mode registers. (*See* Abstract and FIG. 2 of Blomgren.)

Blomgren teaches the emulation of complex CISC instructions using a plurality of RISC instructions, which are referred to as "emulated instructions". As described at col. 7, lines 1-10:

> The instruction decoder 32 is responsible for detecting such instructions, which results in the setting of a RISC mode in loading the emulation routine in memory, such that following execution, the mode register 38 is reset to CISC mode and the instruction pointer is updated to point to the following CISC instruction.

In other words, Blomgren teaches the detection of an emulated CISC instruction and switches to a RISC mode when such an instruction is detected. However, the cited passage above, as well as the entire text of Blomgren is devoid of any reference to detecting whether an input from a plurality of floating point registers includes a token, let along whether a NaTVal token defines an unsuccessful speculative load request.

Furthermore, Blomgren is devoid of any teachings of regarding speculative load requests, or speculative loading of data. In contrast, Blomgren teaches a CISC instruction decoder that detects emulated instructions and signals the unknown opcode, which the Examiner equates to a token, to load pointers with the address of an emulation routine. (*See* cols. 6-7, lines 61-10.)

Accordingly, although Blomgren describes emulation of certain CISC routines with RISC instructions, Applicants respectfully submit that the detection of emulated instructions does not teach NaTVal tokens to indicate an unsuccessful speculative load request, as required by Claim 6 of Group III.

Therefore, a *prima facie* case of anticipation of Claim 6 of Group III is not established and the rejection of Claim 6 should be overturned. Accordingly, Applicants respectfully request that the §102(e) rejection of Claim 6 of Group III be overturned.

E. Group IV: Rejection of Claim 9 as Obvious over Blomgren in View of IEEE

The Examiner rejected all pending claims, including Claim 9 under 103(a) as obvious over Blomgren in view of IEEE.

1. Errors of Law and Fact in the Rejection

The Examiner has made the same errors as described previously with respect to the rejected claims of Groups I-III. In addition, the Examiner has failed to show that the prior art references of Blomgren in view of IEEE teach or suggest all claim features of Claim 9 of Group IV. The Federal Circuit Court of Appeals in In re Rijckaert, 9 F.3d 1531, 28 U.S.P.Q. 2d 1955 (Fed. Cir. 1993) held that:

> In rejecting claims under 35 U.S.C. § 103, the examiner bears the initial burden of presenting a *prima facie* case of obviousness. . . . "A *prima facie* case of obviousness is established when the teaching from the prior art itself would appear to have suggested the claimed subject matter to a person of ordinary skill in the art.". . . If the examiner fails to establish a *prima facie* case, the rejection is

improper and will be overturned. (Emphasis added.) 9 F.3d at 1532, 28 U.S.P.Q. 2d at 1956.

The Examiner has rejected Claim 9 of Group IV under 35 U.S.C. §103(a) as being unpatentable over Blomgren in view of IEEE. The Examiner cites IEEE, which according to the Examiner teaches floating point standards for 32 bit or 64 bit wide instructions. According to the Examiner:

> It would be obvious to one of ordinary skill in the art to use instructions sets of 32 bits and 64 bits because these two instruction sets would be compatible without any outside programmer device written or built to be compatible with the IEEE standards. Therefore, it would be obvious to one of ordinary skill in the art at the time the invention was made to use the instructions with the IEEE standards of 32 bit and 64 bit words in the invention of Blomgren to increase compatibility. Also, the actual word size of the instruction does not matter. *See, In re Rose*. (*See* pp. 8-9, ¶ 24 of the Final Office Action.)

However, even if IEEE disclosed instructions for 32 bit or 64 bit wide ISAs, the Examiner fails to teach that it would be obvious to combine the missing elements provided by IEEE within the teachings of Blomgren.

It is well established that obviousness cannot be established by combining the teachings of the prior art to produce the claimed invention absent the teaching or suggestion supporting such combination. ACS Hospital Sys., Inc. v. Montefiore Hospital, 732 F.2d. 1572, 1577, 221 U.S.P.Q. 929, 933 (Fed. Cir. 1984). Also, one cannot find obviousness through hindsight to construct a claimed invention from elements of the prior art. In re Warner, 379 F.2d 1011, 1016, 154 U.S.P.Q. 173, 177 (C.C.P.A. 1967).

Hence, a *prima facie* case of obviousness of Claim 9 of Group IV has not been established and the rejection of Claim 9 is erroneous and should be overturned.

2. Specific Limitations Not Described in the Prior Art

Claim 9 of Group IV requires a 32 bit size ISA mode and a 64 bit word ISA mode, which as noted by the Examiner is not taught or suggested by Blomgren.

3. Explanation Why Such Limitations Render the Claims Unanticipated by the Prior Art

Here, Blomgren provides no teachings with respect to a multi-mode processor that operates according to a first mode to process instructions from a 32 bit word ISA, referred to herein as a "legacy ISA", as well as a 64 bit word ISA, referred to herein as a "current ISA". As indicated above with reference to the claims of Groups I-III, Blomgren teaches a mechanism for rapid reconfiguration of pipeline alignment between a pipeline optimized for RISC constructions and one optimized for CISC construction, with the use of muxes and mode registers. (*See* Abstract and FIG. 2 of Blomgren.) To this end, Blomgren describes the various mechanisms for

aligning the CISC and RISC pipelines to emulate complex CISC instructions using a plurality of RISC instructions.

Applicants respectfully submit that the entire text of Blomgren is devoid of any teachings with regards to processing instructions from ISAs having different word sizes. Furthermore, as described above with reference to the claims of Group I, the Examiner improperly equates the term "instruction size" with the term "word size". Furthermore, Applicants submit that the combination of Blomgren in view of IEEE, which describes ALU processing of the infinity input encoding, provides no teachings or suggestions with reference to variation in word size between instruction sets processed by a processor, as required by Claim 9 of Group IV.

However, the case law is quite clear in establishing that each and every element of the claim must be exactly disclosed by the combination of references cited by the Examiner. *See, Id.*

Moreover, even if IEEE disclosed instructions for 32 bit or 64 bit wide ISAs, the Examiner fails to teach that it would be obvious to combine the missing elements provided by IEEE within the teachings of Blomgren. Accordingly, Applicants respectfully submit that the features of Claim 9 of Group IV could only be arrived at through inappropriate hindsight.

Therefore, Applicants respectfully submit that a *prima facie* case of obviousness of Claim 9 of Group IV is not established and therefore the rejection of Claim 9 is erroneous and should be overturned. Accordingly, Applicants respectfully request that the §103(a) rejection of Claim 9 of Group IV be overturned.

F. Group V: Rejection of Claims 10-13, 15, 16 and 19 As Anticipated by Blomgren

The Examiner rejected all pending claims, including Claim 10-13, 15, 16 and 19 of Group V under 35 U.S.C. §102(e) as being anticipated by Blomgren.

1. Errors of Law and Fact in the Rejection

The Examiner has made the same errors as previously described with respect to the rejected claims of Group II. In addition, the Examiner has failed to show that each and every element of the claims of Group V are exactly disclosed by Blomgren. Banner Titanium. *Id.*

According to the Examiner, the use of a mode register to indicate the emulation of a CISC routine with RISC instructions, which results in the CISC instruction decoder 32 signaling an unknown opcode from line 40 to mode control logic 30 anticipates the claims of Group V. Applicants respectfully submit that the Examiner has incorrectly associated the term "unknown opcode" with the token as required by independent Claims 10 and 19. (*See* cols. 6-7, lines 61–10.)

Namely, Blomgren teaches the emulation of complex CISC instructions using a plurality of RISC instructions, which are referred to as "emulated instructions".

> The instruction decoder 32 is responsible for detecting such instructions, which results in the setting of an RISC mode and loading of the emulation routine in memory, such that following execution, the mode register 38 is reset to CISC mode and the instruction pointer is updated to point to the following CISC instruction. (Col. 7, lines 1-10.) (Emphasis added.)

Applicants submit that the cited passage provides no reference to whether a token is received as an input to determine what processing to take according to that input, depending on whether the processor is in a first mode or a second mode, as required by independent Claims 10 and 19 of Group V. Hence, a *prima facie* case of anticipation of the claims of Group V has not been established and the rejection of Claims 10-13, 15, 16 and 19 is therefore erroneous.

2. Specific Limitations Not Described in the Prior Art

Each of independent Claims 10 and 19 require:

> fetching an input from at least one of a plurality of floating-point registers;
> detecting whether the input includes a token;
> if the token is detected in the input, checking what mode the processor is in;
> if the processor is in a first mode, processing the input to render an arithmetic result operation;
> if the processor is in a second mode, performing a token specific operation. (Emphasis added.)

3. Explanation Why Such Limitations Render the Claims Unanticipated by the Prior Art

The Applicants claim a method for processing an input to render an arithmetic result if a processor is in a first mode and performing a token specific operation if a processor is in a second mode when a token is detected as an input. In contrast, Blomgren teaches a mechanism for rapid reconfiguration of pipeline alignment between a pipeline optimized for RISC instructions and one optimized for CISC instructions with the use of muxes and mode registers. (*See* Abstract and FIG. 2 of Blomgren.)

Blomgren teaches the emulation of complex CISC instructions using a plurality of RISC instructions, which are referred to as "emulated instructions". As described at col. 7, lines 1-10:

> The instruction decoder 32 is responsible for detecting such instructions, which results in the setting of a RISC mode in loading the emulation routine in memory, such that following execution, the mode register 38 is reset to CISC mode and the instruction pointer is updated to point to the following CISC instruction.

In other words, Blomgren teaches the detection of an emulated CISC instruction and switches to a RISC mode when such an instruction is detected. However, the cited passage above, as well as the entire text of Blomgren is devoid of any reference to detecting whether an input from a plurality of floating point registers includes a token.

Furthermore, Blomgren is devoid of any teachings of performing token specific processing if the processor is in a second mode or processing the input to render an arithmetic result if the processor is in a first mode. In contrast, Blomgren teaches a CISC instruction decoder that detects emulated instructions and signals the unknown opcode, which the Examiner equates to a token, to load pointers with the address of an emulation routine. (*See* cols. 6-7, lines 61-10.)

Accordingly, although Blomgren describes emulation of certain CISC routines with RISC instructions, the processing performed within Blomgren does not vary according to a first or second mode of a multi-mode processor. Applicants respectfully submit that the detection of emulated instructions does not teach the detection of whether an input contains a token, as required by the claims of Group V.

Therefore, a *prima facie* case of anticipation of the claims of Group V is not established and the rejection of Claims 10-13, 15, 16 and 19 should be overturned. Accordingly, Applicants respectfully request that the §102(e) rejection of the claims of Group V be overturned.

G. Group VI: Rejection of Claim 14 As Anticipated by Blomgren

The Examiner rejected all pending claims, including Claim 14 of Group VI under 35 U.S.C. §102(e) as being anticipated by Blomgren.

1. Errors of Law and Fact in the Rejection

The Examiner has made the same errors as previously described with respect to the rejected claims of Groups III and V. In addition, the Examiner has failed to show that each and every element of the claims of Group VI are exactly disclosed by Blomgren. Banner Titanium. *Id.*

According to the Examiner, the use of a mode register to indicate the emulation of a CISC routine with RISC instructions, which results in the CISC instruction decoder 32 signaling an unknown opcode from line 40 to mode control logic 30 anticipates the claims of Group VI. Applicants respectfully submit that the Examiner has incorrectly associated the term "unknown opcode" with the token as required by the claims of Group VI. (*See* cols. 6-7, lines 61–10.)

Namely, Blomgren teaches the emulation of complex CISC instructions using a plurality of RISC instructions, which are referred to as "emulated instructions".

> The instruction decoder 32 is responsible for detecting such instructions, which results in the setting of an RISC mode and loading of the emulation routine in memory, such that following execution, the mode register 38 is reset to CISC mode and the instruction pointer is updated to point to the following CISC instruction. (Col. 7, lines 1-10.) (Emphasis added.)

Applicants submit that the cited passage provides no reference to whether a token received as an input, is an NaTVal token, defines an unsuccessful speculative load request, as required by Claim 14 of Group VI. Hence, a *prima facie* case of anticipation of Claim 14 of Group VI has not been established and the rejection of Claim 14 is therefore erroneous.

2. Specific Limitations Not Described in the Prior Art

Claim 14 of Group VI requires:

> wherein the token represents a "not a thing value" (NaTVal) that defines an unsuccessful speculative load request.

3. Explanation Why Such Limitations Render the Claims Unanticipated by the Prior Art

The Applicants claim a method for performing a token specific operation if a processor is in a second mode and where a NaTVal token defines an unsuccessful, speculative load request. In contrast, Blomgren teaches a mechanism for rapid reconfiguration of pipeline alignment between a pipeline optimized for RISC instructions and one optimized for CISC instructions with the use of muxes and mode registers. (*See* Abstract and FIG. 2 of Blomgren.)

Blomgren teaches the emulation of complex CISC instructions using a plurality of RISC instructions, which are referred to as "emulated instructions". As described at col. 7, lines 1-10:

> The instruction decoder 32 is responsible for detecting such instructions, which results in the setting of a RISC mode in loading the emulation routine in memory, such that following execution, the mode register 38 is reset to CISC mode and the instruction pointer is updated to point to the following CISC instruction.

In other words, Blomgren teaches the detection of an emulated CISC instruction and switches to a RISC mode when such an instruction is detected. However, the cited passage above, as well as the entire text of Blomgren is devoid of any reference to detecting whether an input includes an NaTVal token, which defines an unsuccessful speculative load request.

Furthermore, Blomgren is devoid of any teachings regarding speculative loading of data. In contrast, Blomgren teaches a CISC instruction decoder that detects emulated instructions and signals the unknown opcode, which the Examiner equates to a token, to load pointers with the address of an emulation routine. (*See* cols. 6-7, lines 61-10.)

Accordingly, although Blomgren describes emulation of certain CISC routines with RISC instructions, the processing performed within Blomgren does not provide a teaching or suggestion regarding NaTVal tokens that define an unsuccessful speculative load request. Applicants respectfully submit that the detection of emulated instructions does not teach the detection of NaTVal token that defines an unsuccessful speculative load request, as required by Claim 14 of Group VI.

Therefore, a *prima facie* case of anticipation of Claim 14 of Group VI is not established and the rejection of Claim 14 should be overturned. Accordingly, Applicants respectfully request that the §102(e) rejection of Claim 14 of Group VI be overturned.

H. Group VII: Rejection of Claim 17 as Obvious over Blomgren in View of IEEE

The Examiner rejected all pending claims, including Claim 17 under 103(a) as obvious over Blomgren in view of IEEE.

1. Errors of Law and Fact in the Rejection

The Examiner has made the same errors as described previously with respect to the rejected claims of Groups I, IV and V. In addition, the Examiner has failed to show that the prior art references of Blomgren in view of IEEE teach or suggest all claim features of the claims of Group VII. The Federal Circuit Court of Appeals in In re Rijckaert, 9 F.3d 1531, 28 U.S.P.Q. 2d 1955 (Fed. Cir. 1993) held that:

> In rejecting claims under 35 U.S.C. § 103, the examiner bears the initial burden of presenting a *prima facie* case of obviousness. . . . "A *prima facie* case of obviousness is established when the teaching from the prior art itself would appear to have suggested the claimed subject matter to a person of ordinary skill in the art.". . . If the examiner fails to establish a *prima facie* case, the rejection is improper and will be overturned. (Emphasis added.) 9 F.3d at 1532, 28 U.S.P.Q. 2d at 1956.

The Examiner has rejected Claim 17 of Group VII under 35 U.S.C. §103(a) as being unpatentable over Blomgren in view of IEEE. The Examiner cites IEEE, which according to the Examiner teaches floating point standards for 32 bit or 64 bit wide instructions. According to the Examiner:

> It would be obvious to one of ordinary skill in the art to use instructions sets of 32 bits and 64 bits because these two instruction sets would be compatible without any outside programmer device written or built to be compatible with the IEEE standards. Therefore, it would be obvious to one of ordinary skill in the art at the time the invention was made to use the instructions with the IEEE standards of 32 bit and 64 bit words in the invention of Blomgren to increase compatibility. Also, the actual word size of the instruction does not matter. *See, In re Rose*. (*See* pp. 8-9, ¶ 24 of the Final Office Action.)

However, even if IEEE disclosed instructions for 32 bit or 64 bit wide ISAs, the Examiner fails to teach that it would be obvious to combine the missing elements provided by IEEE within the teachings of Blomgren.

It is well established that obviousness cannot be established by combining the teachings of the prior art to produce the claimed invention absent the teaching or suggestion supporting such combination. ACS Hospital Sys., Inc. v. Montefiore Hospital, 732 F.2d. 1572, 1577, 221 U.S.P.Q. 929, 933 (Fed. Cir. 1984). Also, one cannot find obviousness through hindsight to construct a claimed invention from elements of the prior art. In re Warner, 379 F.2d 1011, 1016, 154 U.S.P.Q. 173, 177 (C.C.P.A. 1967).

Hence, a *prima facie* case of obviousness of the claims of Group VII has not been established and the rejection of Claim 17 is erroneous and should be overturned.

2. Specific Limitations Not Described in the Prior Art

Claim 17 of Group VII requires a 32 bit size ISA mode and a 64 bit word ISA mode, which as noted by the Examiner is not taught or suggested by Blomgren.

3. Explanation Why Such Limitations Render the Claims Unanticipated by the Prior Art

Here, Blomgren provides no teachings with respect to a multi-mode processor that operates according to a first mode to process instructions from a 32 bit word ISA, referred to herein as a "legacy ISA", as well as a 64 bit word ISA, referred to herein as a "current ISA". As indicated above with reference to the claims of Groups I and II, Blomgren teaches a mechanism for rapid reconfiguration of pipeline alignment between a pipeline optimized for RISC constructions and one optimized for CISC construction, with the use of muxes and mode registers. (*See* Abstract and FIG. 2 of Blomgren.) To this end, Blomgren describes the various mechanisms for aligning the CISC and RISC pipelines to emulate complex CISC instructions using a plurality of RISC instructions.

Applicants respectfully submit that the entire text of Blomgren is devoid of any teachings with regards to processing instructions from ISAs having different word sizes. Furthermore, as described above with reference to the claims of Group I, the Examiner improperly equates the term "instruction size" with the term "word size". Furthermore, Applicants submit that the combination of Blomgren in view of IEEE, which describes ALU processing of the infinity input encoding, provides no teachings or suggestions with reference to variation in word size between instruction sets processed by a processor, as required by Claim 17 of Group VII.

However, the case law is quite clear in establishing that each and every element of the claim must be exactly disclosed by the combination of references cited by the Examiner. *See, Id.*

Moreover, even if IEEE disclosed instructions for 32 bit or 64 bit wide ISAs, the Examiner fails to teach that it would be obvious to combine the missing elements provided by IEEE within the teachings of Blomgren. Accordingly, Applicants respectfully submit that the features of Claim 17 of Group VII could only be arrived at through inappropriate hindsight.

Therefore, Applicants respectfully submit that a *prima facie* case of obviousness of Claim 17 of Group VII is not established and therefore the rejection of Claim 17 is erroneous and should be overturned. Accordingly, Applicants respectfully request that the §103(a) rejection of Claim 17 of Group VII be overturned.

IX. CONCLUSION AND RELIEF

Based on the foregoing, Applicant requests that the Board overturn the rejection of all pending claims and hold that all of the claims of the present application are allowable.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN

Dated: April 9, 2004

Joseph Lutz, Reg. No. 43,765

12400 Wilshire Boulevard
Seventh Floor
Los Angeles, California 90025
(310) 207-3800

CERTIFICATE OF MAILING:
I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail, with sufficient postage, in an envelope addressed to: Mail Stop Appeal Brief - Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on April 9, 2004

Marilyn Bass April 9, 2004

Attachment: Exhibit 1

X. APPENDIX

The claims involved in this Appeal are as follows:

1. (Previously Presented) A processor comprising:

a first instruction set engine to process instructions from a first instruction set architecture (ISA) having a first word size;

a second instruction set engine to process instructions from a second ISA having a second word size, the second word size being different than the first word size;

a mode identifier;

a plurality of floating-point registers shared by the first instruction set engine and the second instruction set engine; and

a floating-point unit coupled to the floating-point registers, the floating-point unit processing an input responsive to the mode identifier to produce an output.

2. (Original) The processor of Claim 1 wherein the mode identifier is one of a plurality of bits in a processor status register.

3. (Previously Presented) The processor of Claim 1 wherein the floating-point unit comprises:

pre-processing hardware to detect if a token exists in the input;

an arithmetic unit responsive to the input and the mode identifier; and

post-processing hardware to perform a token specific operation if a token exists in the input.

4. (Previously Presented) The processor of Claim 1 wherein the input includes data stored in at least one of the floating-point registers.

5. (Previously Presented) The processor of Claim 1 wherein the input may contain a token, wherein the floating-point registers are 82 bits wide, and wherein the token being an 82 bit processor known value.

6. (Previously Presented) The processor of Claim 3 wherein the token represents a "not a thing value" (NaTVal) that defines an unsuccessful speculative load request.

7. (Original) The processor of Claim 1 wherein the floating point registers each comprise:

a sign bit,

an exponent; and

a significand.

8. (Original) The processor of Claim 1 wherein the mode identifier indicates whether the processor is in a first mode or a second mode.

9. (Previously Presented) The processor of Claim 1 wherein the mode identifier indicates whether the processor is in a 32 bit word ISA mode or a 64 bit word ISA mode.

10. (Previously Presented) A method in a processor comprising:

fetching an input from at least one of a plurality of floating-point registers;

detecting whether the input includes a token;

if the token is detected in the input, checking what mode the processor is in;

if the processor is in a first mode, processing the input to render an arithmetic result;

if the processor is in a second mode, performing a token specific operation; and

producing an output.

11. (Previously Presented) The method of Claim 10 wherein the input is comprised of at least one operand and at least one operator; wherein detecting comprises examining the at least one operand to determine whether any of the operands correspond to the token; and wherein checking comprises examining a mode identifier to determine whether the processor is in the first mode or the second mode.

12. (Previously Presented) The method of Claim 10 wherein processing comprises executing at least one operation on the at least one operand according to the at least one operator to achieve a result.

13. (Original) The method of Claim 10 wherein performing comprises propagating the token; and wherein producing output comprises setting the output to be the token.

14. (Original) The method of Claim 10 wherein the token represents a "not a thing value" (NaTVal) that defines an unsuccessful speculative load request.

15. (Original) The method of Claim 10 wherein checking comprises checking a mode identifier.

16. (Original) The method of Claim 10 wherein checking comprises checking a mode identifier bit in a processor status register.

17. (Original) The method of Claim 11 wherein the first mode is a 32 bit word ISA mode and the second mode is a 64 bit word ISA mode.

18. (Previously Presented) A multi-mode processor comprising:

a plurality of instruction set engines to process instructions from a plurality of instruction set architectures having different word sizes;

a mode identifier;

a plurality of floating-point registers shared by the instruction set engines; and

a plurality of floating-point units coupled to the floating-point registers, the floating-point units processing an input responsive to the mode identifier.

19. (Previously Presented) A method in a multi-mode processor comprising:

fetching an input from at least one of a plurality of floating-point registers;

detecting whether the input includes at least one token of a plurality of tokens;

if at least one token is detected in the input, checking what mode the processor is in;

processing the input to render an arithmetic result when the processor is in at least a first mode of a plurality of modes; and

performing a token specific operation when the processor is in at least a second mode of a plurality of modes.

Computer Dictionary Online

Medical Dictionary Legal Dictionary

word size search

a b c d e f g h i j k l m n o p q r s t u v w x y z

word size

<processor> The number of bits that a CPU can process at one time. Processors with many different word sizes have existed though powers of two (8, 16, 32, 64) have predominated for many years. A processor's word size is often equal to the width of its external data bus though sometimes the bus is made narrower than the CPU (often half as many bits) to economise on packaging and circuit board costs.

(1995-04-23)

AF/2183 $
2700

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Application No.	09/505,949
Filing Date	February 15, 2000
First Named Inventor	Michael Chow
Art Unit	2183
Examiner Name	Li, Aimee J.
Total Number of Pages in This Submission	29
Attorney Docket Number	42390P6447

ENCLOSURES *(check all that apply)*

- [X] Fee Transmittal Form
  - [X] Fee Attached
- [ ] Amendment / Response
  - [ ] After Final
  - [ ] Affidavits/declaration(s)
- [ ] Extension of Time Request
- [ ] Express Abandonment Request
- [ ] Information Disclosure Statement
  - [ ] PTO/SB/08
- [ ] Certified Copy of Priority Document(s)
- [ ] Response to Missing Parts/ Incomplete Application
  - [ ] Basic Filing Fee
  - [ ] Declaration/POA
- [ ] Response to Missing Parts under 37 CFR 1.52 or 1.53
- [ ] Drawing(s)
- [ ] Licensing-related Papers
- [ ] Petition
- [ ] Petition to Convert a Provisional Application
- [ ] Power of Attorney, Revocation Change of Correspondence Address
- [ ] Terminal Disclaimer
- [ ] Request for Refund
- [ ] CD, Number of CD(s)
- [ ] After Allowance Communication to Group
- [ ] Appeal Communication to Board of Appeals and Interferences
- [X] Appeal Communication to Group **(Appeal Notice, Brief, Reply Brief)**
- [ ] Proprietary Information
- [ ] Status Letter
- [X] Other Enclosure(s) *(please identify below):*

Return Receipt Postcard

Remarks

Appeal Brief submitted in triplicate

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm or Individual name	Joseph Lutz, Reg. No. 43,765 BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP
Signature	
Date	April 9, 2004

CERTIFICATE OF MAILING/TRANSMISSION

I hereby certify that this correspondence is being deposited with the United States Postal Service on the date shown below with sufficient postage as first class mail in an envelope addressed to: Mail Stop Appeal Brief-Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

Typed or printed name	**Marilyn Bass**		
Signature		Date	**April 9, 2004**

Based on PTO/SB/21 (02-04) as modified by Blakely, Sokoloff, Taylor & Zafman (wlr) 02/10/2004.
SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450

FEE TRANSMITTAL for FY 2004

Effective 01/01/2004. Patent fees are subject to annual revision.

☐ Applicant claims small entity status. See 37 CFR 1.27.

TOTAL AMOUNT OF PAYMENT ($) 330.00

Complete if Known	
Application Number	09/505,949
Filing Date	February 15, 2000
First Named Inventor	Michael Chow
Examiner Name	Li, Aimee J.
Art Unit	2183
Attorney Docket No.	42390P6447

METHOD OF PAYMENT *(check all that apply)*

☒ Check ☐ Credit card ☐ Money Order ☐ Other ☐ None

☐ Deposit Account

Deposit Account Number: 02-2666

Deposit Account Name: Blakely, Sokoloff, Taylor & Zafman LLP

The Commissioner is authorized to: *(check all that apply)*

☐ Charge fee(s) indicated below ☒ Credit any overpayments

☒ Charge any additional fee(s) or underpayment of fees as required under 37 CFR §§ 1.16, 1.17, 1.18 and 1.20.

☐ Charge fee(s) indicated below, **except for the filing fee** to the above-identified deposit account

FEE CALCULATION

1. BASIC FILING FEE

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
1001	770	2001	385	Utility filing fee	
1002	340	2002	170	Design filing fee	
1003	530	2003	265	Plant filing fee	
1004	770	2004	385	Reissue filing fee	
1005	160	2005	80	Provisional filing fee	

SUBTOTAL (1) ($)

2. EXTRA CLAIM FEES

		Extra Claims		Fee from below		Fee Paid
Total Claims		- 20** =		X	=	
Independent Claims		- 5 =		X	=	
Multiple Dependent					=	

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description
1202	18	2202	9	Claims in excess of 20
1201	86	2201	43	Independent claims in excess of 3
1203	290	2203	145	Multiple Dependent claim, if not paid
1204	86	2204	43	**Reissue independent claims over original patent
1205	18	2205	9	**Reissue claims in excess of 20 and over original patent

SUBTOTAL (2) ($)

***or number previously paid, if greater, For Reissues, see below*

FEE CALCULATION (continued)

3. ADDITIONAL FEES

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
1051	130	2051	65	Surcharge - late filing fee or oath	
1052	50	2052	25	Surcharge - late provisional filing fee or cover sheet.	
2053	130	2053	130	Non-English specification	
1812	2,520	1812	2,520	For filing a request for *ex parte* reexamination	
1804	920 *	1804	920 *	Requesting publication of SIR prior to Examiner action	
1805	1,840 *	1805	1,840 *	Requesting publication of SIR after Examiner action	
1251	110	2251	55	Extension for reply within first month	
1252	420	2252	210	Extension for reply within second month	
1253	950	2253	475	Extension for reply within third month	
1254	1,480	2254	740	Extension for reply within fourth month	
1255	1,210	2255	605	Extension for reply within fifth month	
1404	330	2401	165	Notice of Appeal	
1402	330	2402	165	Filing a brief in support of an appeal	330.00
1403	290	2403	145	Request for oral hearing	
1451	1,510	2451	1,510	Petition to institute a public use proceeding	
1452	110	2452	55	Petition to revive - unavoidable	
1453	1,330	2453	665	Petition to revive - unintentional	
1501	1,330	2501	665	Utility issue fee (or reissue)	
1502	480	2502	240	Design issue fee	
1503	640	2503	320	Plant issue fee	
1460	130	2460	130	Petitions to the Commissioner	
1807	50	1807	50	Processing fee under 37 CFR 1.17(q)	
1806	180	1806	180	Submission of Information Disclosure Stmt	
8021	40	8021	40	Recording each patent assignment per property (times number of properties)	
1809	770	1809	385	Filing a submission after final rejection (37 CFR § 1.129(a))	
1810	770	2810	385	For each additional invention to be examined (37 CFR § 1.129(b))	
1801	770	2801	385	Request for Continued Examination (RCE)	
1802	900	1802	900	Request for expedited examination of a design application	
Other fee (specify)					

* Reduced by Basic Filing Fee Paid

SUBTOTAL (3) ($) 330.00

SUBMITTED BY

Complete (if applicable)

Name (Print/Type)	Joseph Lutz	Registration No. (Attorney/Agent)	43,765	Telephone	(310) 207-3800
Signature				Date	04/09/04

Based on PTO/SB/17 (10-03) as modified by Blakely, Sokoloff, Taylor & Zafman (wlr) 02/10/2004.
SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450